Data visualization

t-SNE, UMAP

David Hoksza
http://siret.ms.mff.cuni.cz/hoksza

t-SNE

t-Distributed Stochastic Neighbour Embedding

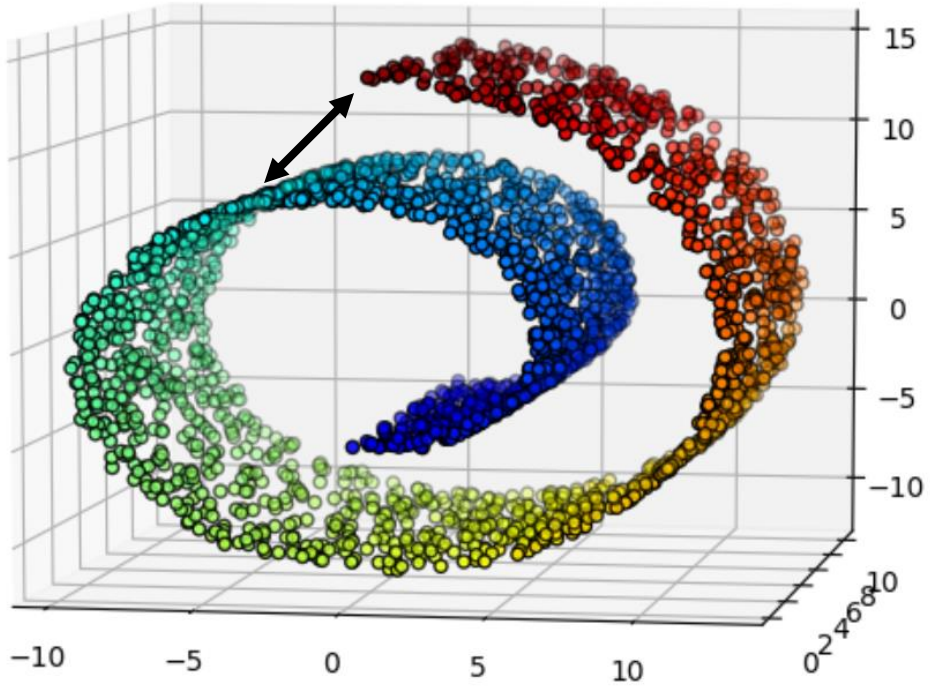
t-Distributed Stochastic Neighbour Embedding (t-SNE)

- Mapping high-dimensional data to low dimensions (dimensionality reduction)
- Focuses on **maintaining local structure** (unlike, e.g., PCA)
  - Global structure preserved by choosing suitable parametrization (perplexity)
- Used for **visualization only** (unlike, e.g., PCA)
- Outliers do not impact t-SNE (unlike, e.g., PCA)
- Builds on top of SNE

(t-)SNE motivation

- **PCA aims at preserving large pairwise distances** because those add most to the variance (minimization of the squared error in the original data)

- In case of data forming **non-linear manifolds, points close to each other in terms of Euclidean distance** can be actually **far apart**



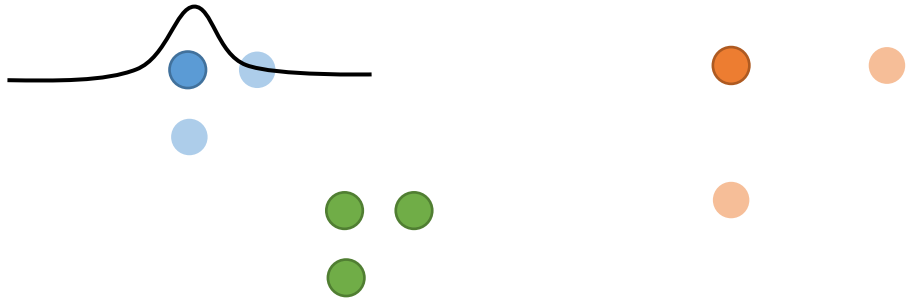
Swiss roll

SNE

1. Models Euclidian distances with **conditional probability-based similarities** (gaussian)

2. Minimizes **difference between similarities** in high- (data) vs low-dimensional (map) data (**Kullback-Leibler divergence**)

3. Uses **gradient descent to minimize** the differences (cost function)



SNE – modeling distances

- **High-dimensional distances** → conditional probabilities
  - $p_{j|i}$
    - similarity of datapoint x_i to x_j
    - the conditional probability, that x_i would pick x_j as its neighbor if neighbors were picked in proportion to their probability density under a **Gaussian centered at x_i with variance σ_i** (controlled by perplexity)
  - $p_{i|i} = 0$

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}$$

Independence on the point's density

- **Low-dimensional distances** → conditional probabilities
  - Variance set to $\frac{1}{\sqrt{2}}$
  - $q_{i|i} = 0$

$$q_{j|i} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq i} \exp\left(-\|y_i - y_k\|^2\right)}$$

- Good mapping would have $p_{j|i}$ and $q_{j|i}$ equal for every j

Kullback-Leibler (KL) divergence

- Measures information **difference between two distributions** p and q
  - E.g., real, complex distribution of data (p) vs simple, approximating distribution (q)

- Based on entropy
  - Number of bits needed to encode our data
  - Entropy of a distribution p

$$H = -\sum_{i=1}^{N} p(x_i) \log p(x_i)$$

- **KL divergence**

$$D_{KL}(p||q) = \sum_{i=1}^{N} p(x_i) \left(\log p(x_i) - \log q(x_i)\right) = \sum_{i=1}^{N} p(x_i) \left(\log \frac{p(x_i)}{q(x_i)}\right)$$

  - Expectation (*střední hodnota*) of the log difference between the probability of data in p vs q
  - In case of approximating distribution: **how much information we expect to lose** (bits if base of log is 2) if using q instead of p

SNE - KL divergence

- **Cost function** in SNE **modeled by KL divergence**

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}},$$

  - P_i - conditional probability distribution over all other datapoints given x_i
  - Q_i - conditional probability distribution over all other map points given y_i

  - KL asymmetric → **preserving local structure**
    - large cost for using widely separated map points to represent nearby datapoints
    - small cost for using nearby map points to represent widely separated datapoints

SNE - perplexity

- We do not want to have the same variance for each datapoint
- Choice of **variance driven by perplexity** (global parameter)
  - With **growing variance**, the **entropy decreases**

$$Perp(P_i) = 2^{H(P_i)}$$

where $H(P_i)$ is the Shannon entropy of P_i in bits

$$H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i}$$

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}$$

  - Balances attention between **local (low perplexity) and global aspects (high perplexity)** of data
  - **Typical values between 5 and 50** (see t-SNE plots analysis slides)

SNE – cost function optimization

- We aim at **minimization of C** (sum of KL divergence over all points)

- **Gradient descent (GD)** → gradient of C

$$\frac{\delta C}{\delta y_i} = 2\sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j)$$

  - Initialized by sampling map points randomly from a Gaussian centered around the origin with small variance
  - GD with momentum to speed up the convergence
  - After each iteration, a gaussian noise is added to the map points with gradually reducing variance of the noise (kind of simulated annealing)

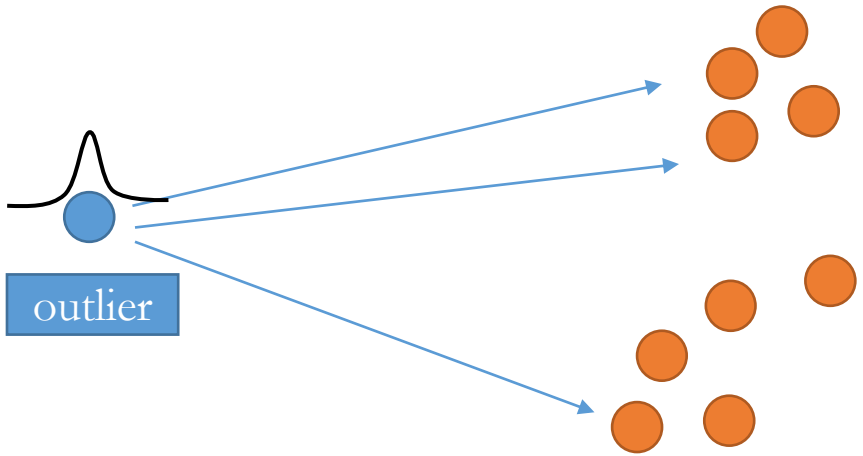
t-SNE

- Replaces SNE with **symmetric SNE**
  - Fixes issue with **outliers**
  - **Speeds up** convergence

- Replaces **gaussian for modeling low-dimensional points with Student's t-distribution** with a single degree of freedom (longer tails)
  - Fixes **crowding problem**

Outliers

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)} \qquad q_{j|i} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq i} \exp\left(-\|y_i - y_k\|^2\right)}$$



$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}},$$

$p_{j|i}$ extremely small $\forall j \Rightarrow$ location of the low-dim projection has very little effect on the cost function $\Rightarrow$ position not well determined

Symmetric SNE (1)

- Models joint distributions → optimizes s**ingle KL divergence instead of sum of KLs**

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

  - i.e., $p_{ij} = p_{ji}$ and $q_{ij} = q_{ji}$

In the low-dim space, the similarities are modeled as

$$q_{ij} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq l} \exp\left(-\|y_k - y_l\|^2\right)}$$

In the high-dim space, the similarities are modeled as

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

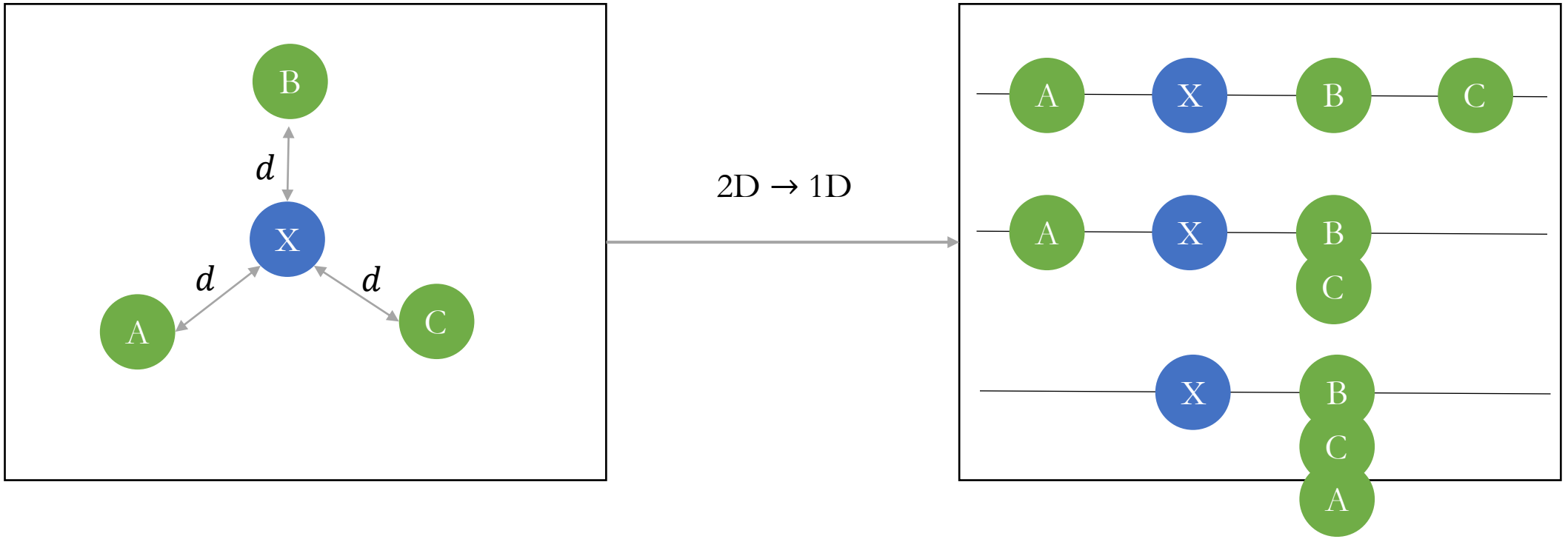
Ensures that $\sum_j p_{ij} > \frac{1}{2n}$ for all x_i → each x_i makes a significant contribution to the cost function

Crowding issue

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

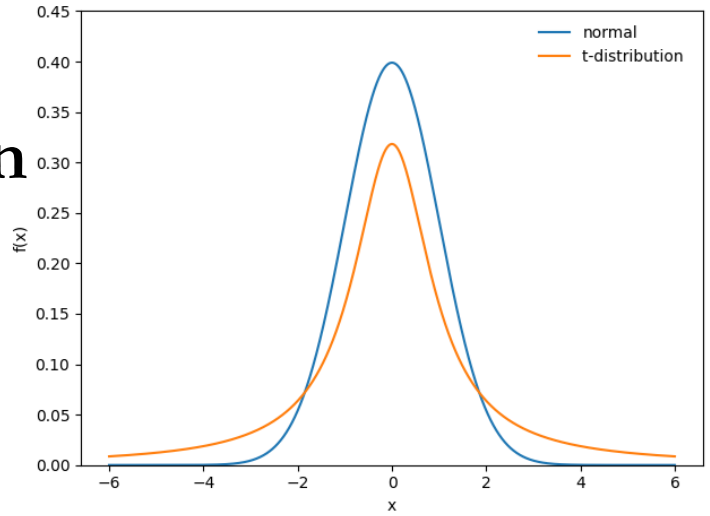
$$q_{ij} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq l} \exp\left(-\|y_k - y_l\|^2\right)}$$

- Moderately equidistant high-dim points tend to get squashed on a single point → crowding



t-Distribution

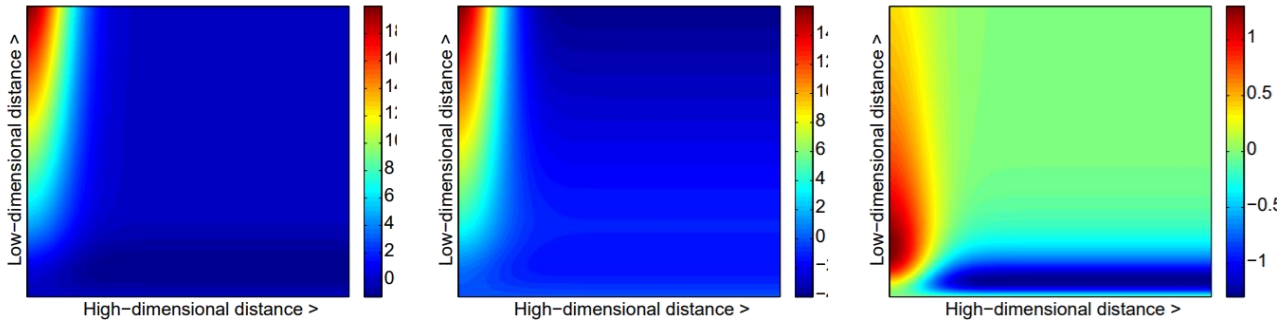
- In **low-dimensional** space **t-SNE uses t-distribution** which has heavier tails allowing moderate distances in high-dim, to be modeled by larger distance in low-dim



$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$$

- Resulting in gradient providing repulsion for too close points in low-dim space

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j) \left(1 + \|y_i - y_j\|^2\right)^{-1}$$

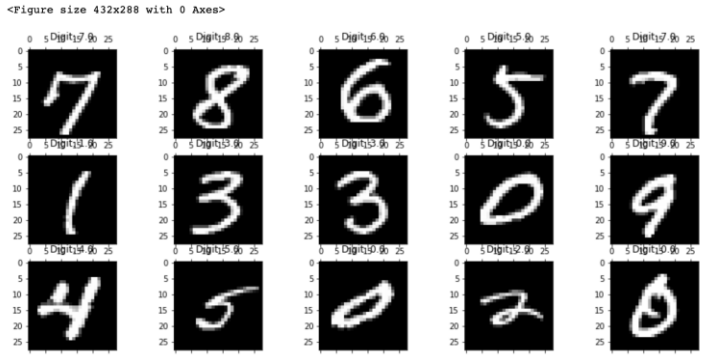


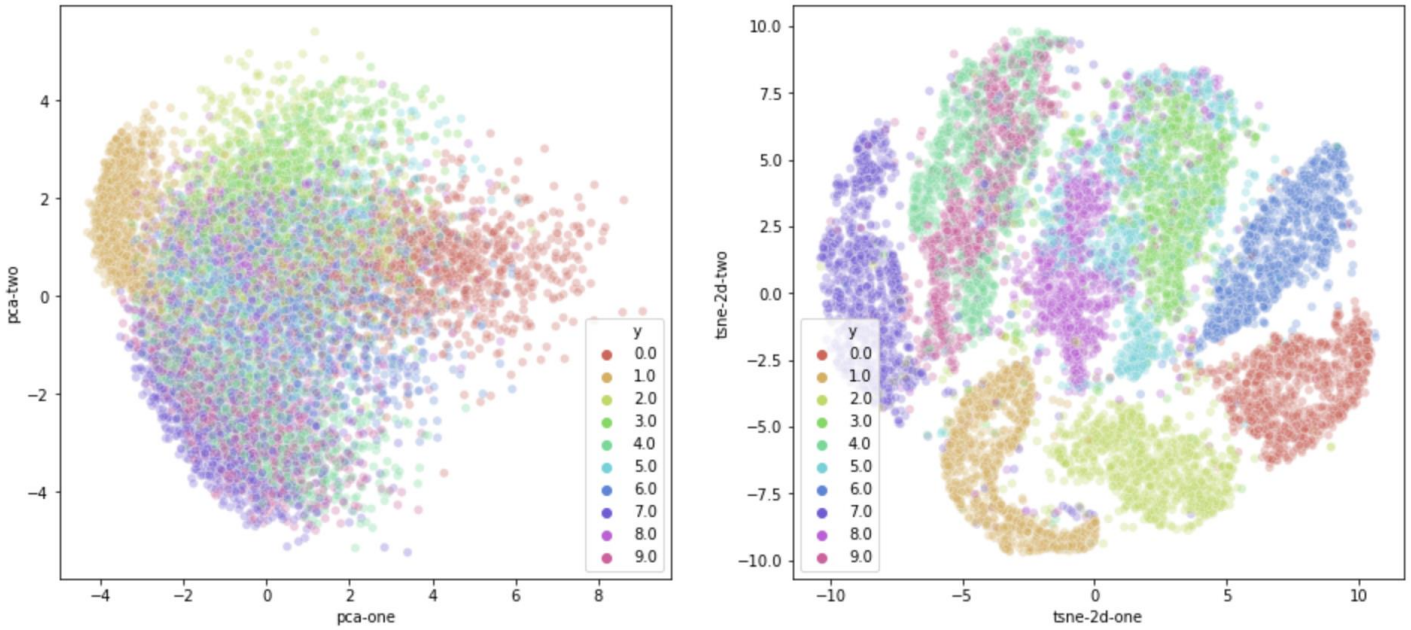
(a) Gradient of SNE. (b) Gradient of UNI-SNE. (c) Gradient of t-SNE.

Figure 1: Gradients of three types of SNE as a function of the pairwise Euclidean distance between two points in the high-dimensional and the pairwise distance between the points in the low-dimensional data representation.

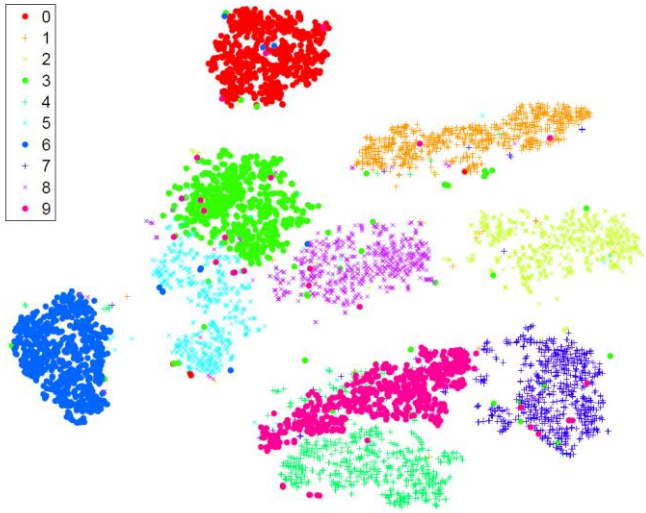
source: Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.11 (2008).

Example on MNIST dataset

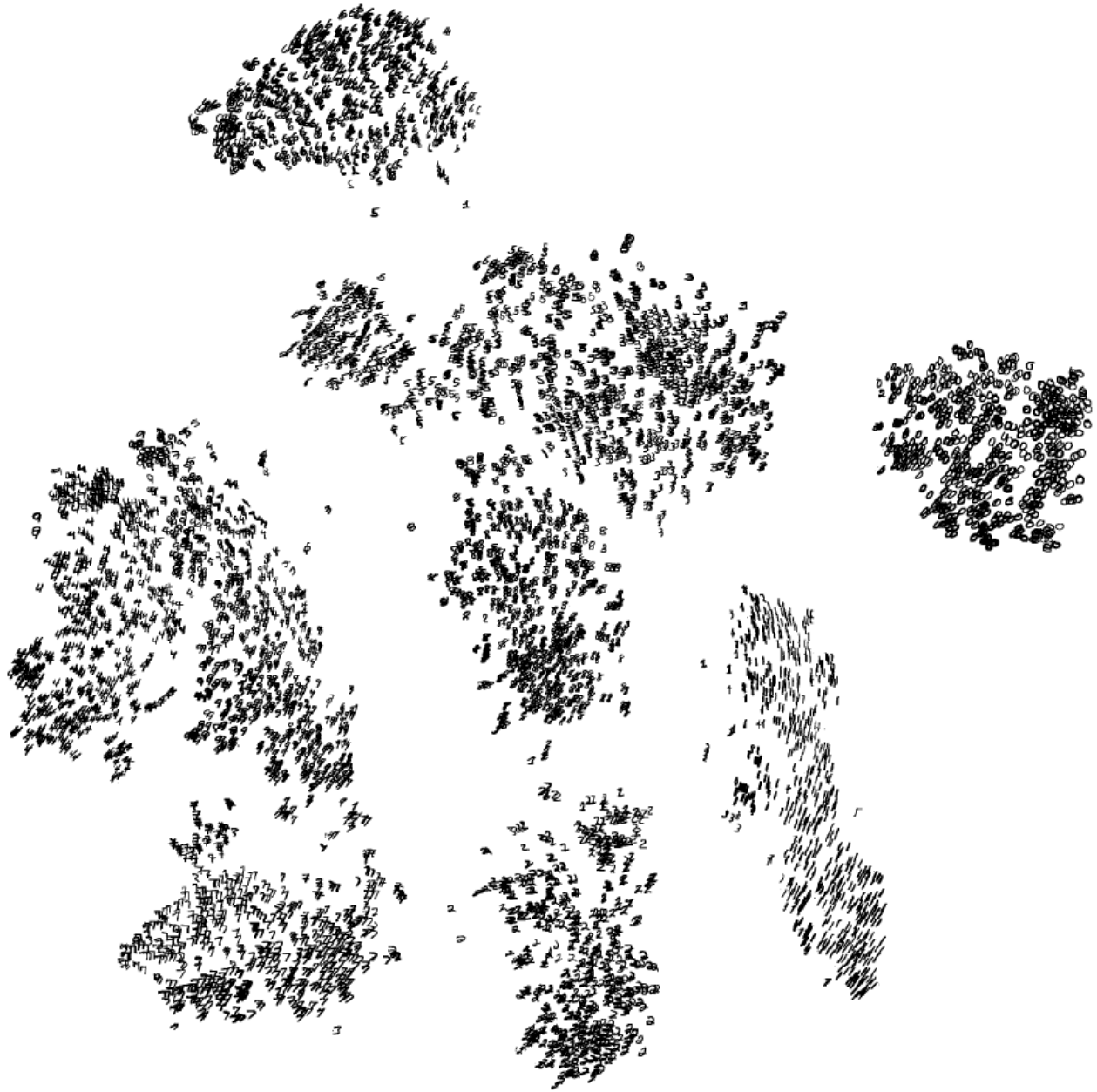




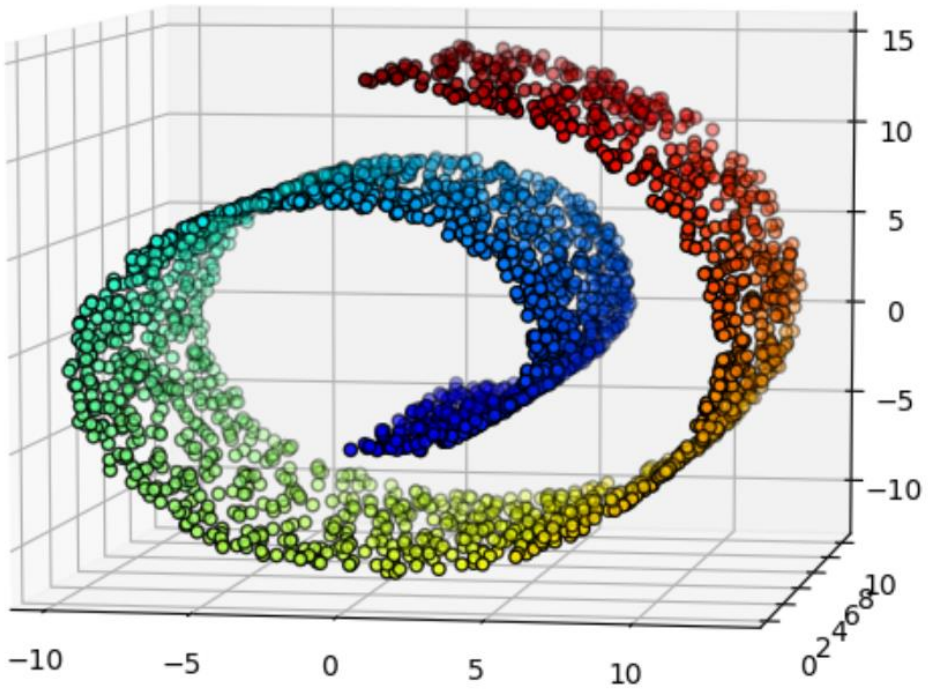
source: https://towardsdatascience.com/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b

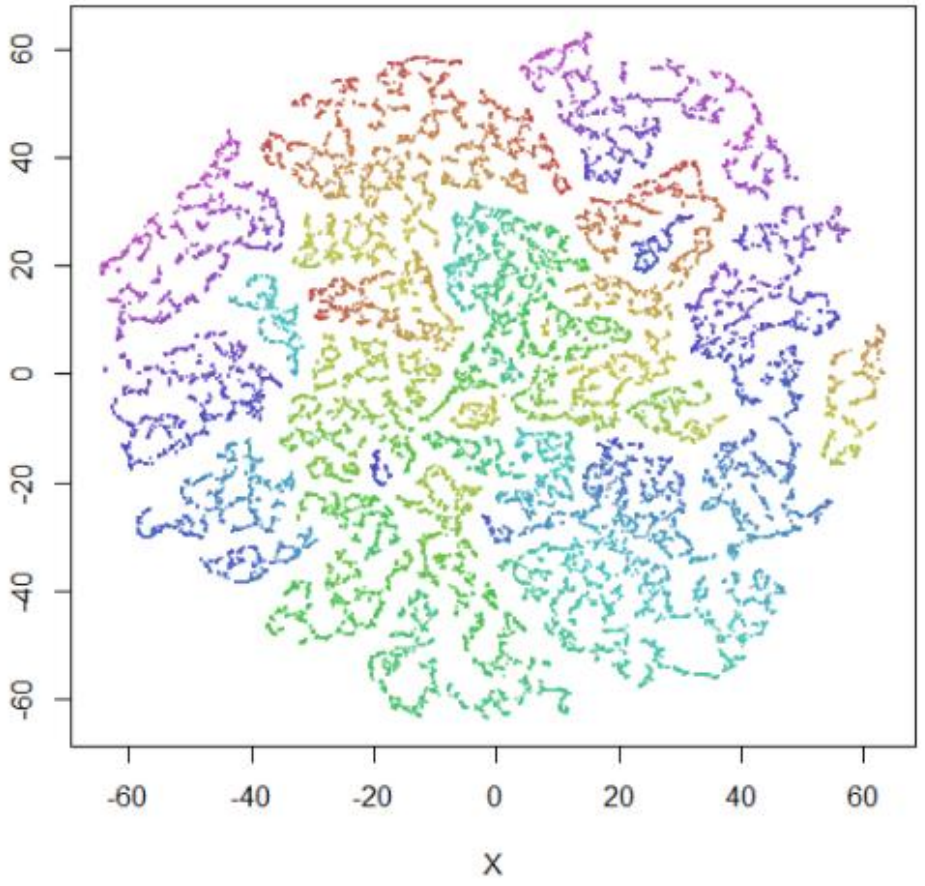


source: L.J.P. van der Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9(Nov):2579-2605, 2008.



Swiss roll example

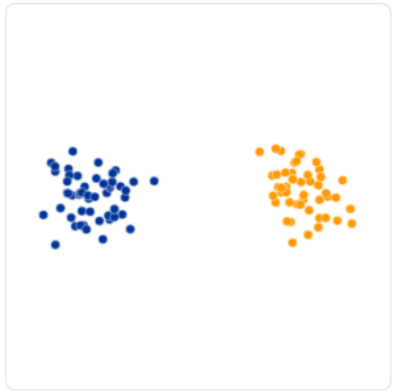




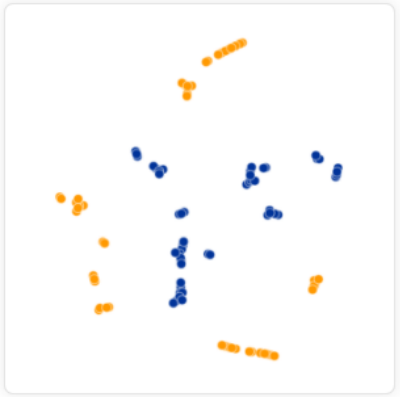
source: https://jlmelville.github.io/uwot/umap-examples.html#swiss_roll

Effect of perplexity

Perplexity should by < number of data points

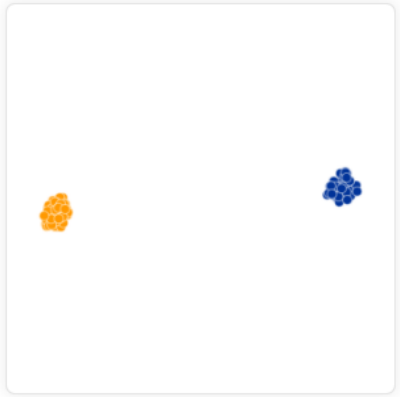


Original



Perplexity: 2
Step: 5,000

Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000

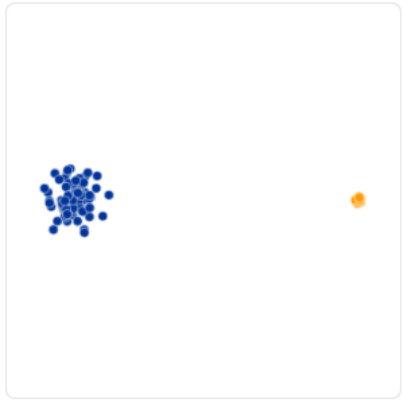
Perplexity: 50
Step: 5,000

Perplexity: 100
Step: 5,000

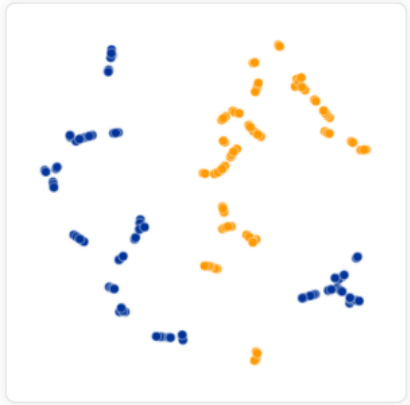
source: https://distill.pub/2016/misread-tsne/

Effect of perplexity

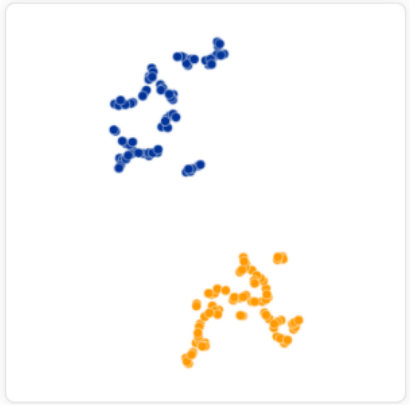
Cluster sizes mean nothing



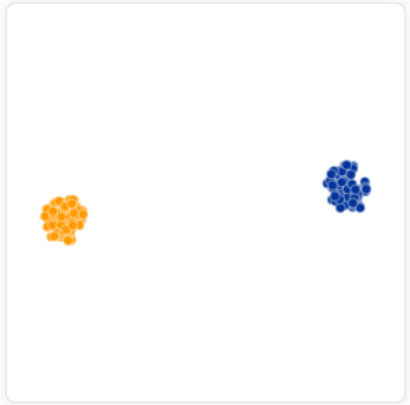
Original



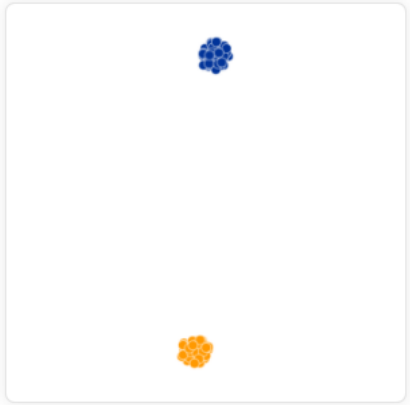
Perplexity: 2
Step: 5,000



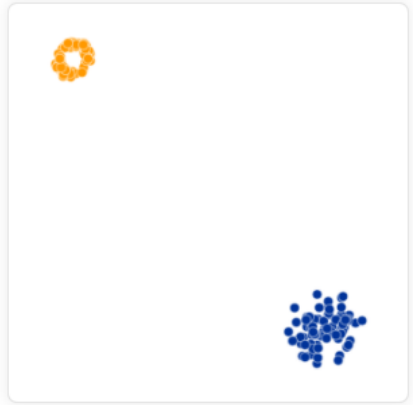
Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000



Perplexity: 50
Step: 5,000

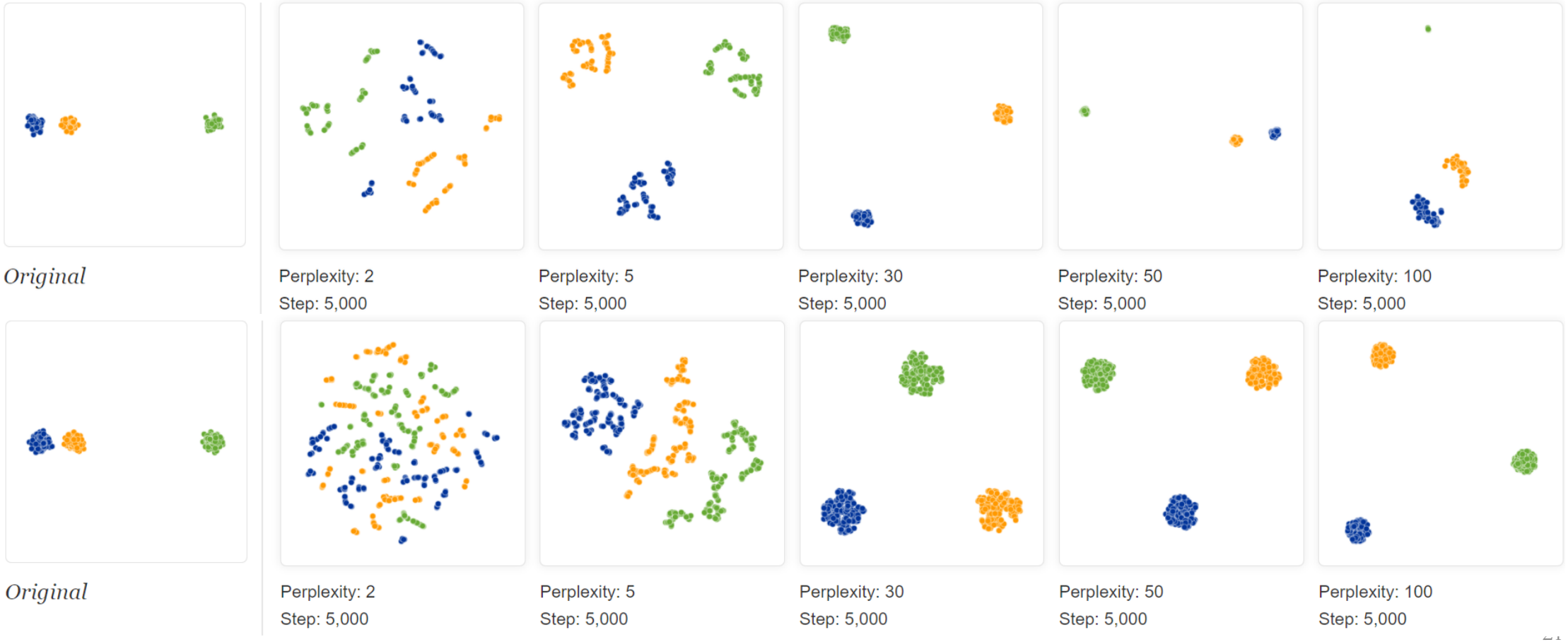


Perplexity: 100
Step: 5,000

source: https://distill.pub/2016/misread-tsne/

Effect of perplexity

Distances between well-separated clusters are not easily reproducible without fine-tuning perplexity

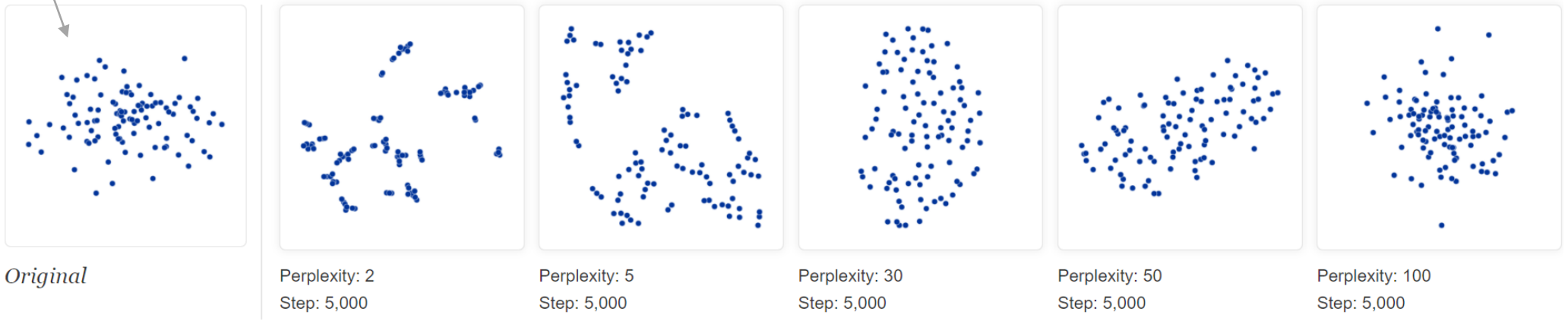


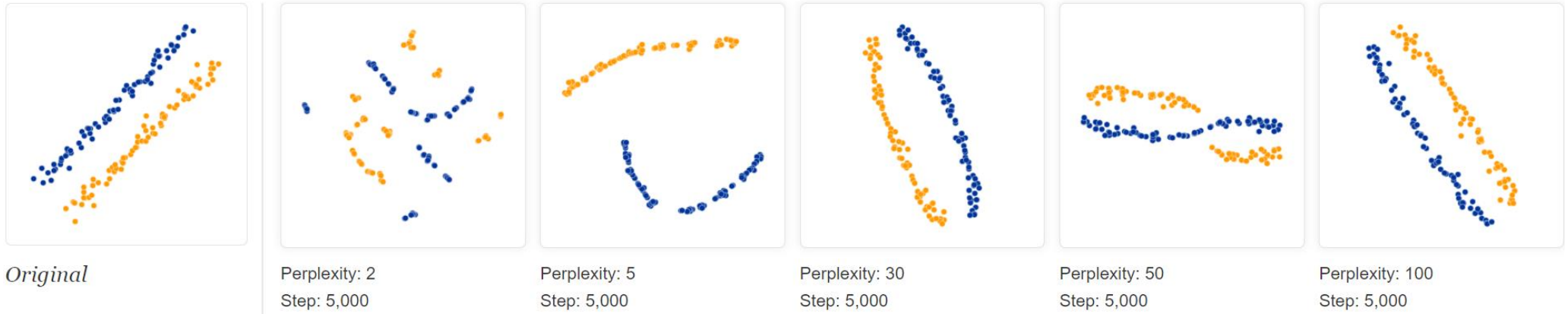
source: https://distill.pub/2016/misread-tsne/

Effect of perplexity

50 dim, standard deviation in coordinate i is $1/i$, 1st and 2nd PCs

Shapes are preserved with suitable perplexity

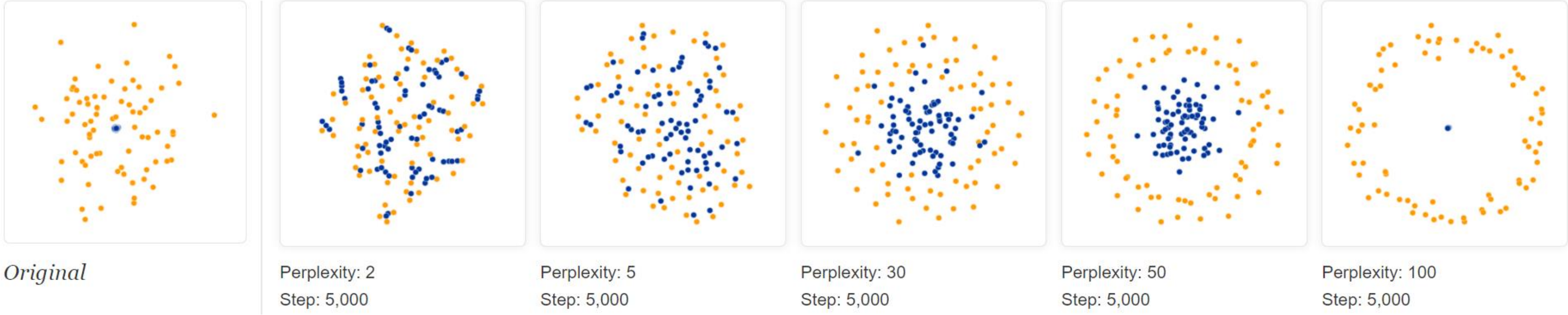


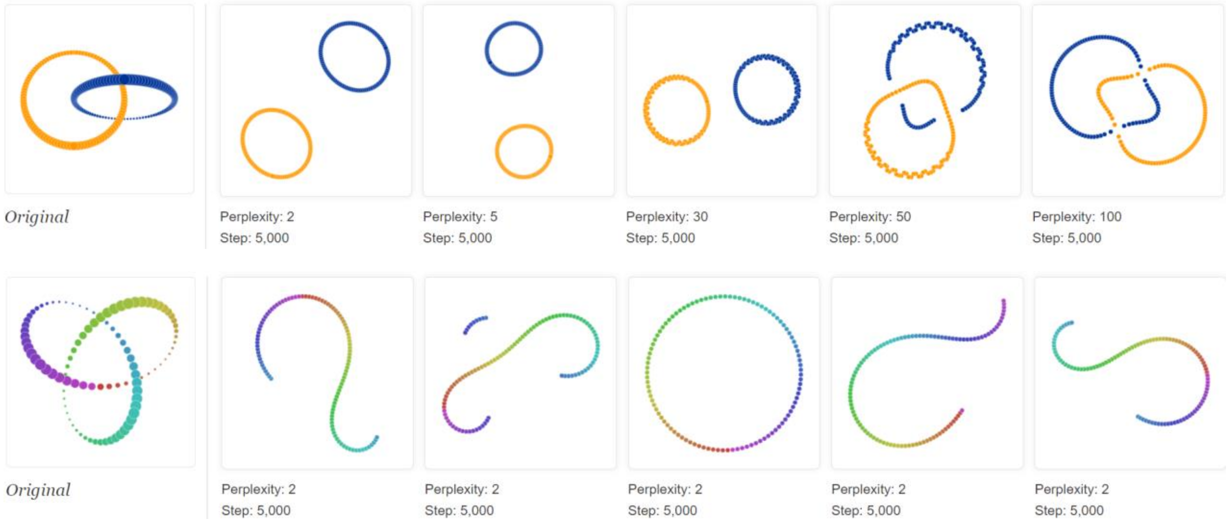


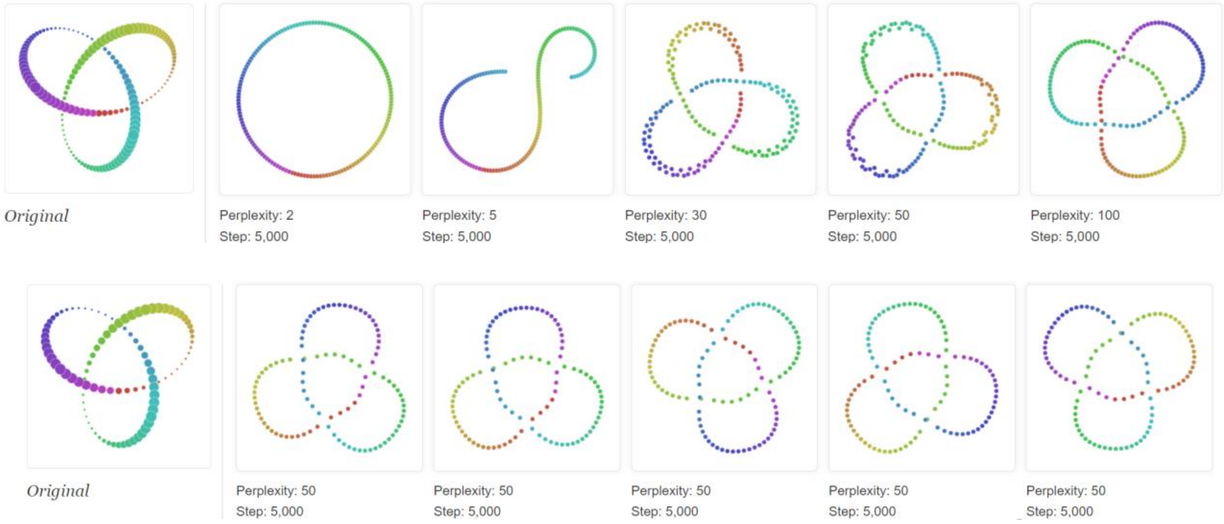
source: https://distill.pub/2016/misread-tsne/

Effect of perplexity

Topology can be preserved with suitable perplexity value







source: https://distill.pub/2016/misread-tsne/

UMAP

Uniform Manifold Approximation and Projection for Dimension Reduction

UMAP

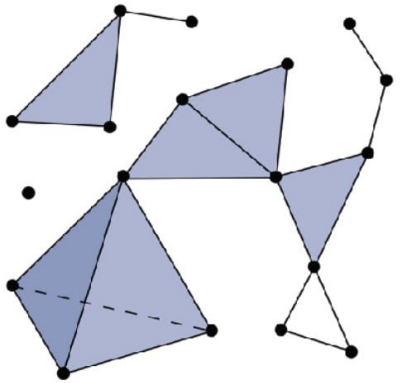
- Rooted in **manifold theory and topological** data **analysis**
  - **Presumption** that **data lie on a manifold** embedded in a high-dim space which we **want to detect and project** to low-dim
  - **Representation** of high-dim and low-dim data with **k-NN graphs**

- Preserves (better than t-SNE] distances between clusters → **preservation of global structure**

- **Faster** than t-SNE

UMAP

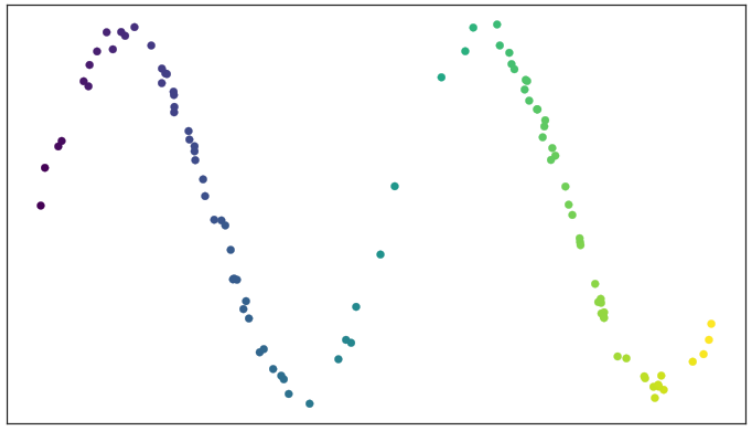
- Construction of a **weighted graph in high dimensions**
  - Combinatorial representation of the underlying topology (it's convex hull) using simplicial complexes → cover of the space



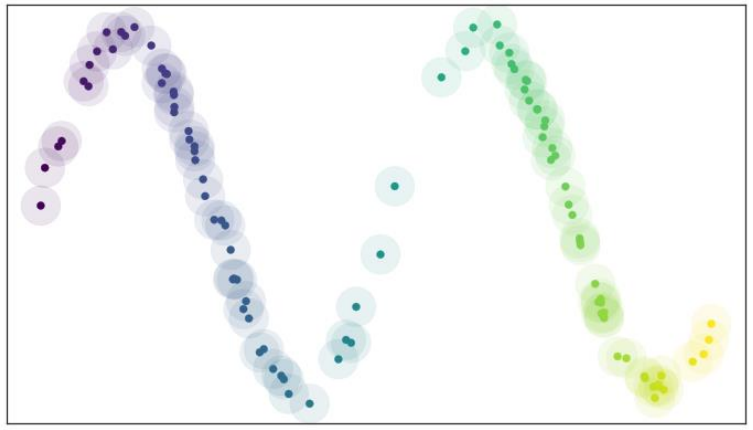
simplicial complex

- Finding the **most appropriate layout in lower dimensions**

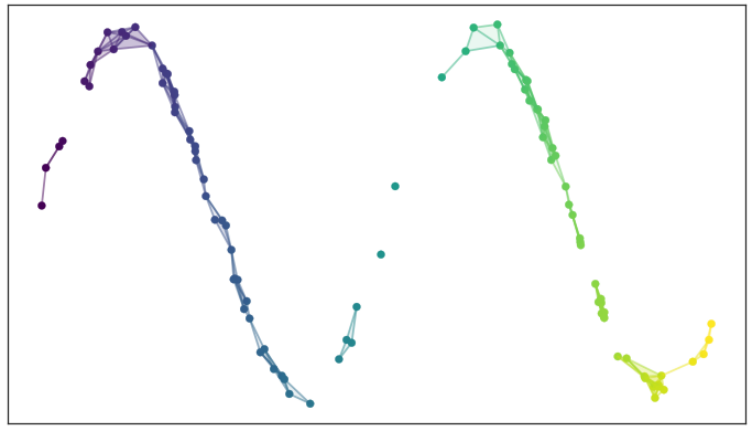
High-dim representation



Data are considered as samples from a continuous manifold



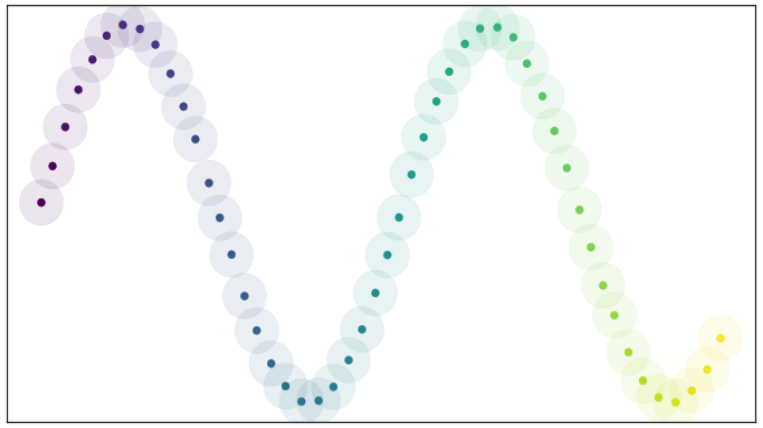
A cover of the manifold formed by open balls placed on each data point



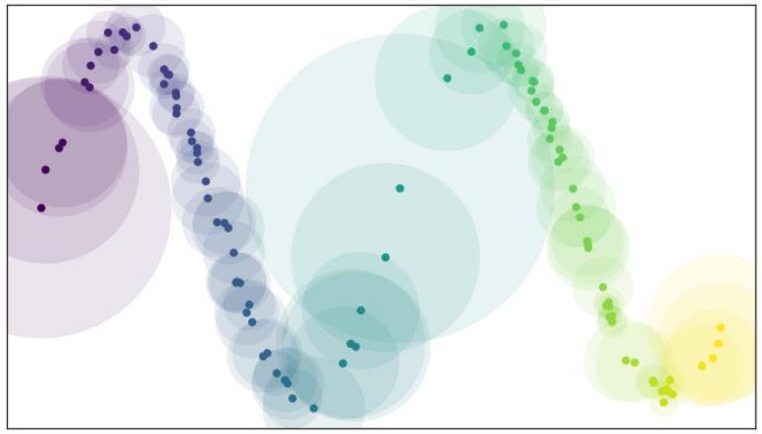
A simplicial complex of the cover (Nerve of a covering) formed by 0- and 1-simplices.

source: https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

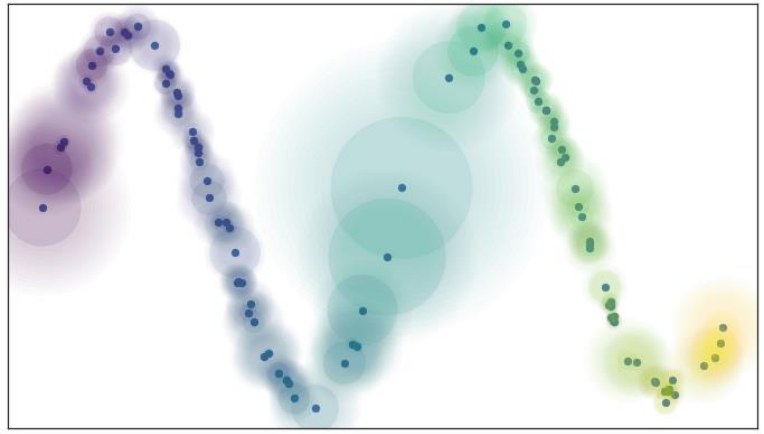
High-dim representation - choosing the radius



If the data were uniformly distributed on the manifold, the cover would be good

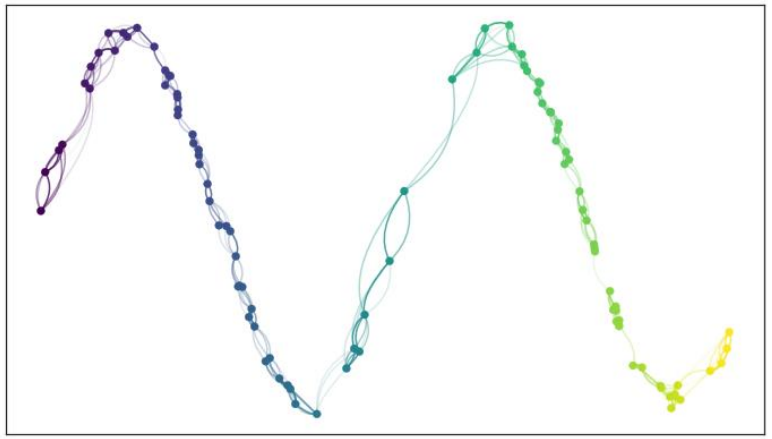


Varying the notion of distance on the manifold makes the uniform assumption true. The balls above has all the same size with respect to local distance on the manifold. This can be done by defining Riemannian metric on the manifold

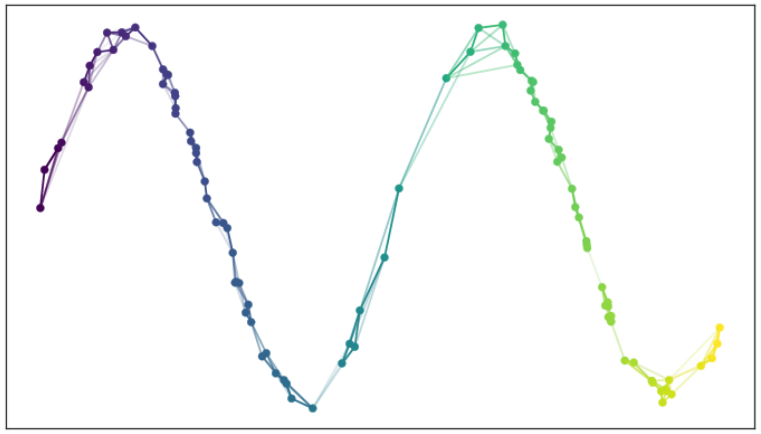


Generalization of the cover to fuzzy cover, i.e., a point is in the neighborhood with a probability (fuzzy balls).
Moreover, for the manifold to be locally connected, i.e., not to have isolated points → 1NN will always be

High-dim representation – graph



Simplicial complexes of the fuzzy covering have 1-simplices with a given probability (based on the local distance) → weighted graph representation.



However, for each pair of points we may have up to 2 edges where $\mathrm{d}(a \to b) \neq \mathrm{d}(b \to a)$. Under a probabilistic fuzzy union the combination of weights on the edges is given by $f(\alpha, \beta) = \alpha + \beta - \alpha \,.\, \beta$

Practical construction of the high-dim graph

hyper-parameter

- Compute k nearest neighbors for each point

- Compute distance to nearest neighbor ρ_i and σ_i

$$\rho_i = \min\{d(x_i, x_{i_j}) \mid 1 \leq j \leq k, d(x_i, x_{i_j}) > 0\}$$

$$\sum_{j=1}^{k} \exp\left(\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right) = \log_2(k)$$

- Define weighted directed graph $G = (X, E, w)$

$$E = \{(x_i, x_{i_j}) \mid 1 \leq j \leq k, 1 \leq i \leq N\}$$

$$w((x_i, x_{i_j})) = \exp\left(\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right)$$

Low-dim representation

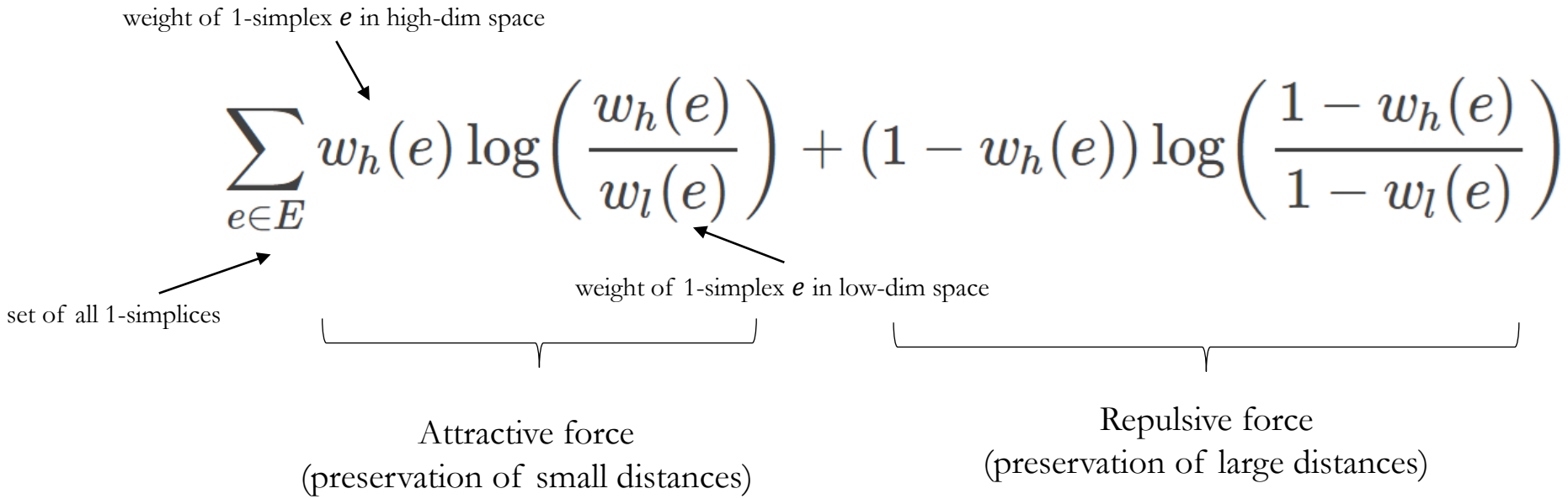
- In **low-dim the manifold** on which the data should lie is the low-dim **Euclidean space** we are embedding to → no varying notion of distance across the manifold

- We need to set the correct nearest neighbor distance for the local connectivity → hyper-parameter **min_dist**
  - controlling how tightly points are clumped together in the resulting layout

The low-dimensional representation is thus a graph in 2D space with edge weights derived from the minimum distance between points min_dist

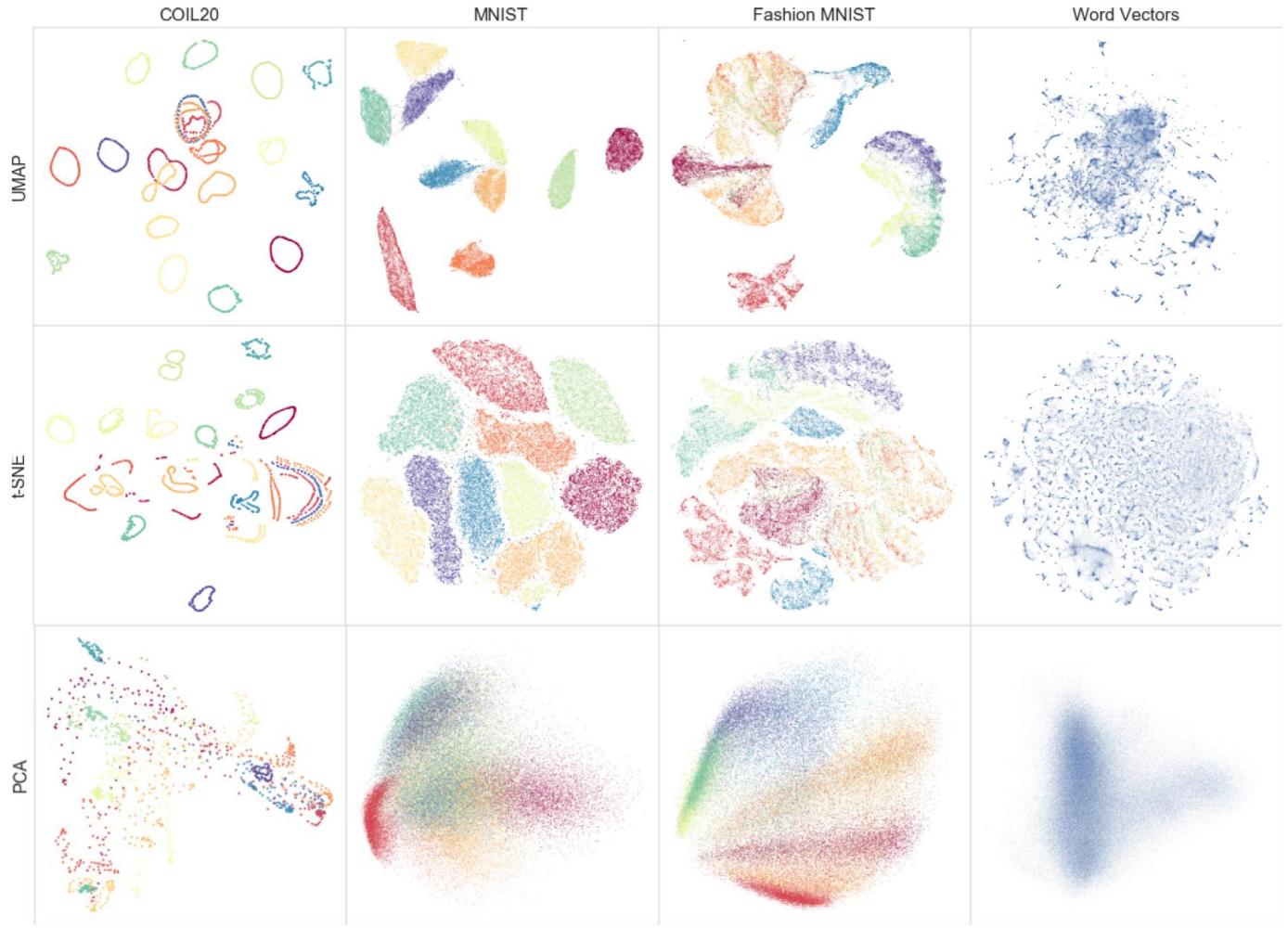
Finding good low-dim representation

- Minimization of cross-entropy



Optimization via SGD + negative sampling

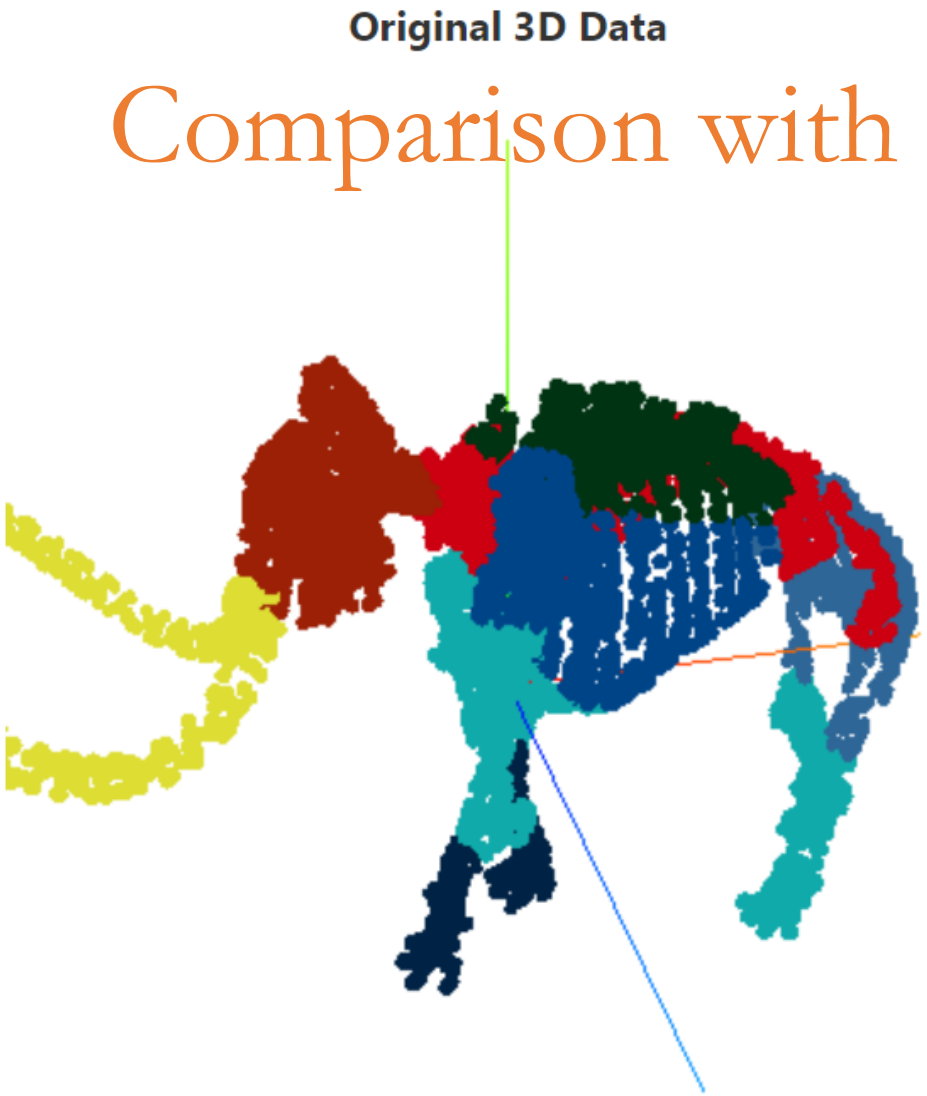
Comparison on standard datasets

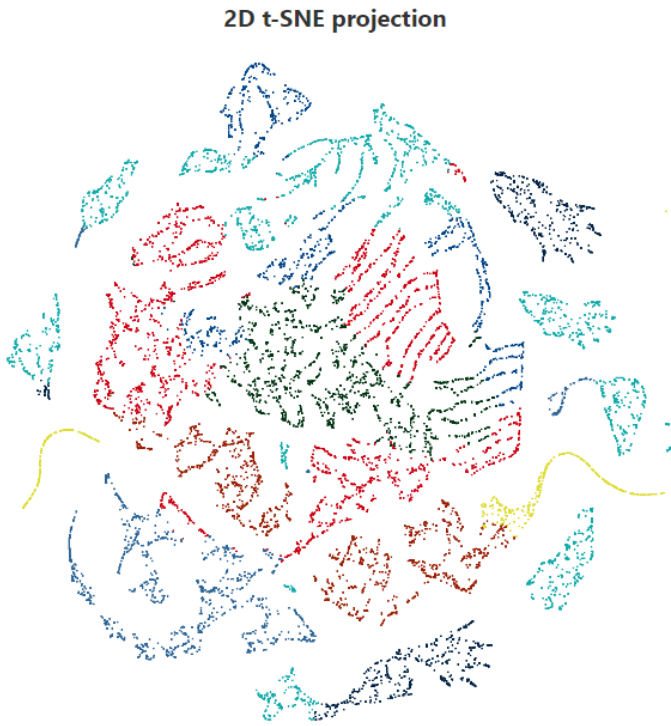


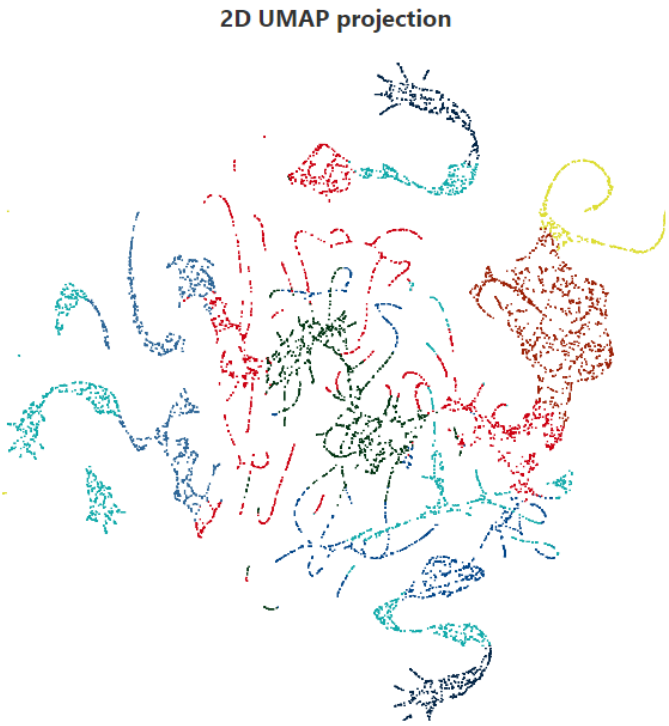
source: McInnes, Leland, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction." arXiv preprint arXiv:1802.03426 (2018)

https://jlmelville.github.io/uwot/umap-examples.html

Comparison with t-SNE on 3D mammoth





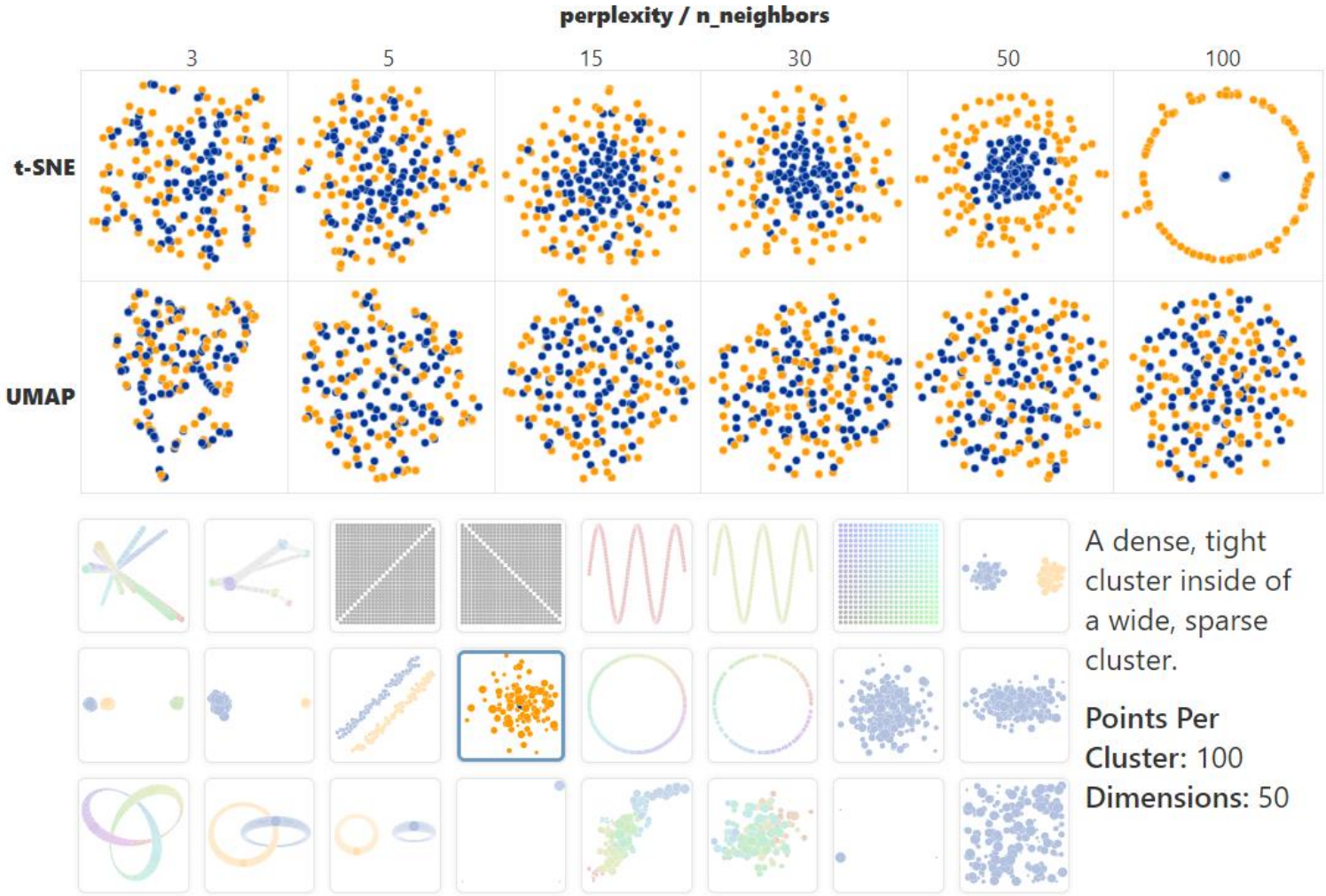


perplexity: 100
time: 16m 1s

n_neighbors: 15
min_dist: 0.1
time: 1m 2s

https://pair-code.github.io/understanding-umap/

Comparison on toy datasets

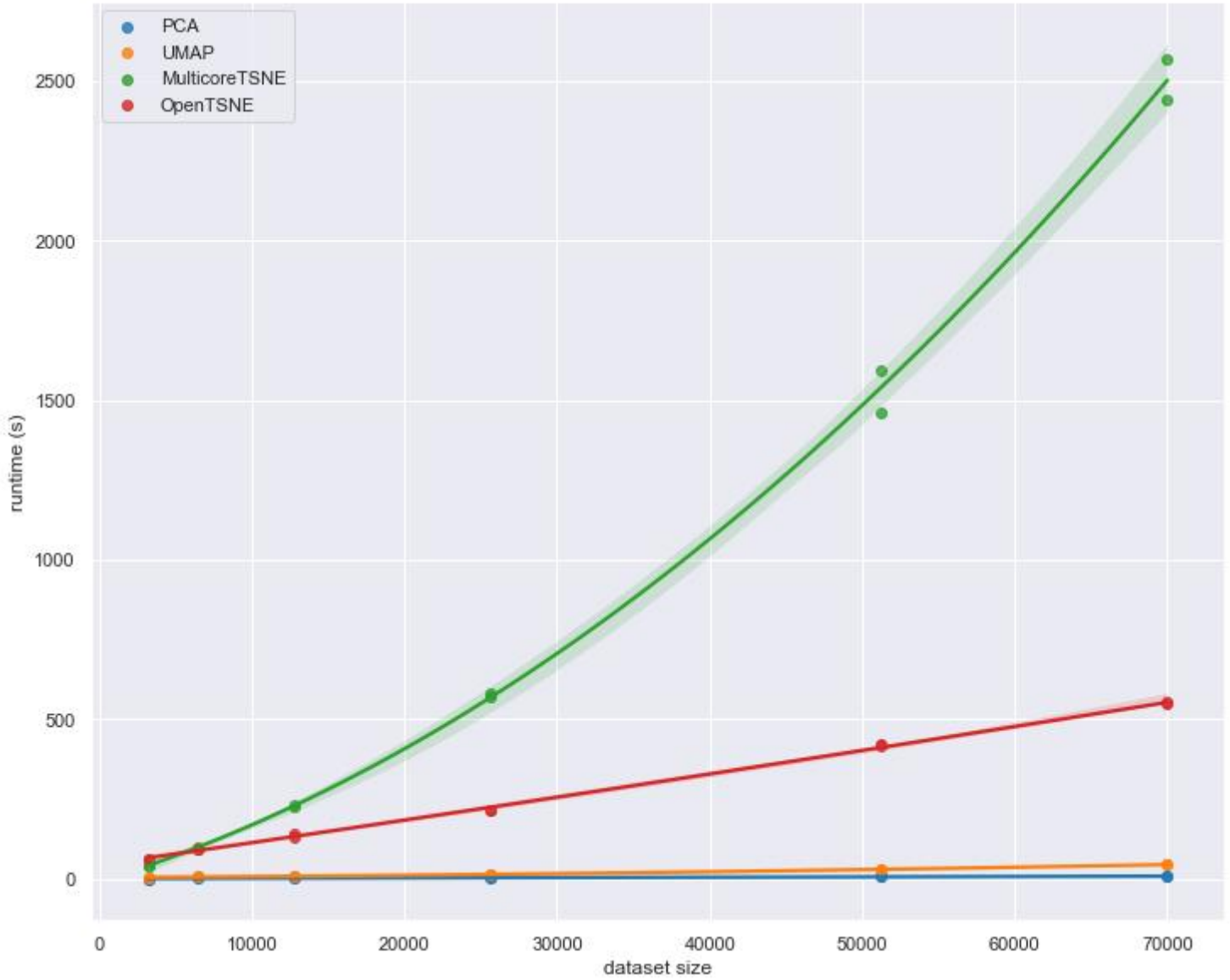


https://pair-code.github.io/understanding-umap/

Interpretation

- Similar observations as in case of t-SNE
- Cluster sizes should not be interpreted
- Distances between cluster should not be interpreted
- Random noise might not look random

Speed



source: https://umap-learn.readthedocs.io/en/latest/performance.html

Sources

- Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.11 (2008).
- https://distill.pub/2016/misread-tsne/

- McInnes, Leland, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction." arXiv preprint arXiv:1802.03426 (2018).
- https://umap-learn.readthedocs.io/
- https://pair-code.github.io/understanding-umap/